B Het installeren en testen van versiebeheersysteem Git

Volg de stap voor stap instructies voor het installeren en testen van git:

Stap 1

Open de terminal(mac) of commandline(windows).

Voer **terminal** of **cmd** in via de zoekfunctie van je OS (Operating System).

Stap 2

Download en Installeer Git voor Windows of Mac

(Windows: https://git-scm.com/download/win)

(Mac: https://sourceforge.net/projects/git-osx-installer/)

Stap 3

Check of Git goed geïnstalleerd is met het **git --version** commando.

Als het goed is zie je het versienummer van git verschijnen

Stap 4

Navigeer met de *terminal* of *CMD* naar de map waarbinnen je Python code (helloWorld.py) staat:

a. Gebruik het commando **cd..** (win) of **cd ..** (mac) om uit de huidige map te gaan. (omhoog)
b. Gebruik **dir** of **ls** om te kijken welke mappen en bestanden de huidige locatie bevat.
c. Gebruik **cd naamVanMap** om een map in te gaan.

Stap 5

Maak met het versiebeheersysteem Git een nieuwe *"Repository"* aan:

a. Maak een nieuwe "Repository" aan op je computer. Dit zorgt ervoor dat Git alles bijhoud wat er gebeurt in de map waarin je code staat. Doe dit met het commando **git init.**

   Git heeft nu een onzichtbare map ".git" aangemaakt in de map waarin je je bevind

b. Controleer of de map **"/.git"** bestaat met het commando **dir /a**

   Als het goed is zie je hem in de lijst staan

Stap 6

Voeg bestanden toe aan de *"tracker"* van Git. Het systeem zal alle wijzigingen aan deze bestanden gaan volgen.

a. Gebruik het commando **git status**

*Je zal nu zien dat je **helloWorld.py** bestand in het rood wordt weergegeven als "untracked file"*

b. Gebruik het commando **git add helloWorld.py** om het bestand toe te voegen aan de lijst van bestanden die door git "getracked" moeten worden.

*Je kunt in plaats van het bovenstaande commando ook **git add -A** gebruiken om in 1x alle bestanden in de map te volgen*

c. Gebruik het commando **git status**

Als het goed is zie je nu dat de file helloWorld.py in het groen wordt weergegeven met de melding "no changes to commit"

Stap 7

Maak aanpassingen en bewaar deze als een "commit". Commits zijn een soort tussenstations voor de staat van je documenten. Je kunt dus altijd weer terug naar die versie.

a. Maak een aanpassing in de tekst die door je programma **helloWorld.py** wordt weergegeven.
b. Check via de "interpreter" of je code nog werkt. Gebruik het commando **python helloWorld.py**
c. Gebruik **git status**

Als het goed is staat er nu in het groen Modified: helloWorld.py. Dit komt omdat je een aanpassing aan de file hebt gemaakt en deze nog niet hebt gecommit.

d. Laat Git een commit maken met het commando **git commit -m "omschrijving"**

*Hierbij bedenk je zelf een goede omschrijving maakt van wat je hebt veranderd. In dit geval zou de omschrijving: "aanpassing van geprinte tekst" een goede omschrijving zijn. Let op dat het invoeren van een **commit message** (omschrijving) verplicht is. Je moet immers terug kunnen lezen welke wijzigingen er zijn gedaan in een commit.*

e. Gebruik **git status**

Als het goed is zie je nu "nothing to commit, working tree is clean". Alle veranderingen binnen je getrackte documenten zijn nu netjes gecommit. Dit is een goede en veilige situatie waarbij je later altijd weer terug kunt naar de huidige staat van je documenten.

f. Gebruik het commando **git log** om een overzicht te geven van je gemaakte commits.

Laat met je Log aan de docent zien dat je een commit hebt gemaakt.